# REPORT DOCUMENTATION PAGE

1

Form Approved OMB NO. 0704-0188



| 1. REPORT DATE (DD-MM-YYYY) | 2. REPORT TYPE | 3. DATES COVERED (From - To) |
|---|---|---|
| | New Reprint | - |

**4. TITLE AND SUBTITLE**

ML-PDA and ML-PMHT: Comparing Multistatic Sonar Trackers for VLO Targets Using a New Multitarget Implementation

**5a. CONTRACT NUMBER**

W911NF-10-1-0369

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

611102

**6. AUTHORS**

Steven Schoenecker, , Peter Willett, Fellow, Yaakov Bar-Shalom,

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAMES AND ADDRESSES**

University of Connecticut - Storrs
438 Whitney Road Ext., Unit 1133

Storrs, CT 06269 -1133

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)**

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

**10. SPONSOR/MONITOR'S ACRONYM(S)**

ARO

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

57823-CS.66

**12. DISTRIBUTION AVAILIBILITY STATEMENT**

Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

**14. ABSTRACT**

The maximum-likelihood probabilistic data association (ML–PDA) tracker and themaximum-likelihood probabilistic multihypothesis (ML–PMHT) tracker are tested in their capacity as algorithms for very low observable (VLO) targets (meaning 6-dB postsignal processing or even less) and are then applied to five synthetic benchmark multistatic active sonar scenarios featuring

**15. SUBJECT TERMS**

VLO Targets,Multitarget Trackers

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 15. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Yaakov Bar-Shalom |
| UU | UU | UU | UU | | 19b. TELEPHONE NUMBER 860-486-4823 |

Standard Form 298 (Rev 8/98)
Prescribed by ANSI Std. Z39.18

**Report Title**

ML-PDA and ML-PMHT: Comparing Multistatic Sonar Trackers for VLO Targets Using a New Multitarget Implementation


**ABSTRACT**

The maximum-likelihood probabilistic data association (ML–PDA) tracker and themaximum-likelihood probabilistic multihypothesis (ML–PMHT) tracker are tested in their capacity as algorithms for very low observable (VLO) targets (meaning 6-dB postsignal processing or even less) and are then applied to five synthetic benchmark multistatic active sonar scenarios featuring multiple targets, multiple sources, and multiple receivers. Both methods end up performing well in situations where there is a single target or widely spaced targets. However, ML–PMHT has an inherent advantage over ML–PDA in that its likelihood ratio (LR) has a simple multitarget formulation, which allows it to be implemented as a true multitarget tracker. This formulation, presented here for the first time, gives ML–PMHT superior performance for instances where multiple targets are closely spaced with similar motion dynamics.

# REPORT DOCUMENTATION PAGE (SF298)
# (Continuation Sheet)

Continuation for Block 13

ARO Report Number 57823.66-CS
ML-PDA and ML-PMHT: Comparing Multistatic S...

Block 13: Supplementary Note


Approved for public release; distribution is unlimited.

# ML–PDA and ML–PMHT: Comparing Multistatic Sonar Trackers for VLO Targets Using a New Multitarget Implementation

Steven Schoenecker, Peter Willett, *Fellow, IEEE*, and Yaakov Bar-Shalom, *Fellow, IEEE*

***Abstract*—The maximum-likelihood probabilistic data association (ML–PDA) tracker and the maximum-likelihood probabilistic multihypothesis (ML–PMHT) tracker are tested in their capacity as algorithms for very low observable (VLO) targets (meaning 6-dB postsignal processing or even less) and are then applied to five synthetic benchmark multistatic active sonar scenarios featuring multiple targets, multiple sources, and multiple receivers. Both methods end up performing well in situations where there is a single target or widely spaced targets. However, ML–PMHT has an inherent advantage over ML–PDA in that its likelihood ratio (LR) has a simple multitarget formulation, which allows it to be implemented as a true multitarget tracker. This formulation, presented here for the first time, gives ML–PMHT superior performance for instances where multiple targets are closely spaced with similar motion dynamics.**

***Index Terms*—Bistatic, expectation maximization (EM), low observable, maximum likelihood, maximum-likelihood probabilistic data association (ML–PDA), maximum-likelihood probabilistic multihypothesis (ML–PMHT), multistatic, multitarget, multitarget ML–PMHT, sonar, tracking.**

## I. Introduction

THE maximum-likelihood probabilistic data association (ML–PDA) tracker and the maximum-likelihood probabilistic multihypothesis (ML–PMHT) tracker are both simple, straightforward algorithms that can be used in an active multistatic sonar framework. With some basic assumptions about a target (or targets) as well as the environment, likelihood ratios (LRs) can be developed for both algorithms and then optimized. The main difference between the two algorithms is in the measurement-to-target assignment model; ML–PDA assumes that at most one measurement per scan can originate from a target, while ML–PMHT allows for any number of measurements to have originated from a target. While this assumption may reduce the appeal of ML–PMHT to some, the resulting algorithm does offer advantages in both its implementation (especially fine-scale optimization) and in terms of its multitarget formulation.

The main contributions of this work are as follows. First, we theoretically develop and implement ML–PMHT as a true multitarget tracker. As part of this, we find an expression for the Cramér–Rao lower bound (CRLB) for ML–PMHT and show that ML–PMHT seems to be a *statistically efficient* estimator. As a result, this CRLB can be used as a covariance estimate for ML–PMHT. Moving on to Monte Carlo testing, we provide an example of the capability of the algorithms as very low observable (VLO) trackers. [In this work, we use "low observable" to refer to targets with a signal-to-noise ratio (SNR) between 6 and 12 dB and "very low observable" to refer to targets with an SNR of 6 dB or less.] We then demonstrate that while ML–PMHT and ML–PDA have identical performance in single-target cases, ML–PMHT is better than ML–PDA in multitarget cases or when more than one measurement per scan is produced by the target.

The paper has the following structure. Section II briefly reviews the ML–PDA and ML–PMHT (single-target) log-likelihood ratios (a much more extensive discussion of this can be found in [1] and [2]). Section III develops the novel multitarget ML–PMHT (including the CRLB for ML–PMHT). Next, Section IV describes in detail the simulator used to create data for Monte Carlo runs on the multistatic sonar scenarios considered; the five different scenarios used in this work are presented here as well. Finally, Section V presents the results of the Monte Carlo testing. We note that portions of this paper have appeared in [1] and [2]; this paper is an expansion of these works. These previous works provided preliminary comparisons between ML–PDA and ML–PMHT when they were both implemented in a sequential single-target mode.

## II. Single-Target ML–PDA Versus ML–PMHT

This section briefly describes the theory of ML–PDA and ML–PMHT (single-target) log-likelihood ratios (LLRs). It then shows how, under certain conditions, the ML–PDA LLR and the ML–PMHT LLR should converge to the same expression. If such conditions are met, there should be no performance difference between the two algorithms.

Manuscript received February 29, 2012; revised August 31, 2012; accepted February 16, 2013. Date of publication May 16, 2013; date of current version April 10, 2014. This work was supported by the U.S. Office of Naval Research (ONR) under Grants N00014-10-10412 and N00014-10-1-0029, and the U.S. Army Research Office (ARO) under Grant W911NF-06-1-0467.

**Associate Editor: D. A. Abraham**.

S. Schoenecker is with the Sensors and Sonar Department, Naval Undersea Warfare Center, Newport, RI 02841 USA and also with the Department of Computer and Electrical Engineering, University of Connecticut, Storrs, CT 06269 USA (e-mail: steven.schoenecker@navy.mil).

P. Willett and Y. Bar-Shalom are with the Department of Computer and Electrical Engineering, University of Connecticut, Storrs, CT 06269 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JOE.2013.2248534

### A. Likelihood Ratios

The ML–PDA concept was initially developed in [3]. Subsequently, in [4]–[6], it was applied in a multistatic active application, which is how we currently employ it. The assumptions used to develop the ML–PDA LR are [7], [8] as follows.

- A single target is present in each frame with known detection probability $P_d$. Detections are independent across frames.
- There are zero or one measurements per frame from the target.
- The kinematics of the target are deterministic. The motion is usually parameterized as a straight line, although any other parameterizations (e.g., ballistic motion) can be used.
- False detections (clutter) are uniformly distributed in the search volume and their number is Poisson distributed with known density.
- Amplitudes of target and false detections are Rayleigh distributed. The parameter of each Rayleigh distribution is known (although the SNR may be tracked [9] in the case that it is not known; then $P_d$ becomes time varying, which is easy to accommodate).
- Target measurements are corrupted with zero-mean Gaussian noise with known variance.
- Measurements at different times, conditioned on the parameterized state, are independent.

The assumptions for ML–PMHT are almost all the same as these, with one significant difference. Instead of allowing only zero or one measurement per scan to originate from the target, the ML–PMHT target-assignment model relaxes this constraint and allows any number of measurements to originate from the target. The development of the ML–PDA and ML–PMHT LRs has been presented in detail in [2]; we just summarize the results here.

The ML–PDA LLR for a batch of data is

$$\Lambda(\mathbf{x}, Z) = \sum_{i=1}^{N_w} \ln \left\{ 1 - P_d + \frac{P_d}{\lambda} \sum_{j=1}^{m_i} p[\mathbf{z}_j(i) \,|\, \mathbf{x}] \rho_j(i) \right\}. \quad (1)$$

In this equation, $P_d$ is the target probability of detection in a scan, $\lambda$ is the spatial clutter density, $N_w$ is the number of scans, and $m_i$ is the number of measurements in the $i$th scan. Next, $\mathbf{z}_j(i)$ is the $j$th measurement in the $i$th scan, $p[\mathbf{z}_j(i) \,|\, \mathbf{x}]$ is a target-centered Gaussian, and $\rho_j(i)$ is the amplitude LR. Finally, $\mathbf{x}$ is the target state, and $Z$ is the (entire) set of measurements in the batch.

The ML–PMHT LLR is given by

$$\Lambda'(\mathbf{x}, Z) = \sum_{i=1}^{N_w} \sum_{j=1}^{m_i} \ln\{\pi_0 + \pi_1 V p[\mathbf{z}_j(i) \,|\, \mathbf{x}] \rho_j(i)\} \quad (2)$$

where $\pi_0$ is the prior probability that a measurement is from clutter, $\pi_1$ is the prior probability that a measurement is from the target, and $V$ is the search volume in the measurement space. Note that the ML–PDA LR is a product of a large sum (based on the total probability theorem with respect to all the joint association events) while the ML–PMHT LR is a double product of the sum of two terms (based on the total probability theorem for each measurement separately, since each could be target or clutter originated, and these events are assumed independent across measurements). This difference is the key in 1) allowing the computation of the ML–PMHT CRLB in a much easier manner than for the ML–PDA CRLB; and 2) allowing the extension of the ML–PMHT for multiple targets.

### B. Relationship Between ML–PDA and ML–PMHT

At first glance, the LLRs for ML–PDA and ML–PMHT appear very dissimilar. However, the terms in the LLR for ML–PDA are actually just a subset of the terms in the LR for ML–PMHT. Consider the ML–PMHT likelihood (not log-likelihood) ratio for a single scan. This is written as

$$\frac{p_1[Z(i) \,|\, \mathbf{x}]}{p_0[Z(i) \,|\, \mathbf{x}]} = \prod_{j=1}^{m_i} \{\pi_0 + \pi_1 V p[\mathbf{z}_j(i) \,|\, \mathbf{x}] \rho_j(i)\}. \quad (3)$$

For any arbitrary number of measurements $m_i$, this equation multiplied out yields

$$\begin{aligned} \frac{p_1[Z(i) \,|\, \mathbf{x}]}{p_0[Z(i) \,|\, \mathbf{x}]} = {} & \pi_0^{m_i} + \pi_0^{m_i - 1} \pi_1 V \sum_{j=1}^{m_i} \mathcal{N}_j \\ & + \pi_0^{m_i - 2} \pi_1^2 V^2 \sum_{j_1=1}^{m_i - 1} \sum_{j_2 > j_1}^{m_i} \mathcal{N}_{j_1} \mathcal{N}_{j_2} + \cdots \\ & + \pi_1^{m_i} V^{m_i} \prod_{j=1}^{m_i} \mathcal{N}_j \end{aligned} \quad (4)$$

where the terms $\mathcal{N}_j$ represent $p[\mathbf{z}_j(i) \,|\, \mathbf{x}]$, the target-centered Gaussian terms. It should also be noted that the amplitude distribution ratios have been left out of the above equation; including them would not affect the following discussion. Equation (4) is the ML–PMHT LR multiplied out for a single scan. It allows for more than one measurement to be assigned to a target; the terms with products of target-centered Gaussians account for this case. In the case of ML–PDA, the target assignment model allows only zero or one measurement to be assigned to the target. The first term in (4) accounts for the case of zero measurements being assigned to the target, and the second term in the equation accounts for the case of one measurement being assigned to the target. The other terms account for two or more measurements being assigned to the target. If the ML–PDA measurement model holds (or is at least very close to holding), then the product of the probabilities of two (or more) measurements being from a target is essentially zero and the terms with a product of two or more target-centered Gaussians in the above expression go to zero. As a result, just the first and second terms are left. In this case, we can claim the following relationships. First

$$1 - P_d = \pi_0^{m_i}. \quad (5)$$

Then

$$P_d = m_i \pi_0^{m_i - 1} \pi_1 \quad (6)$$

and, since $E[m_i] = \lambda V$, one can take (approximately)

$$\pi_0^{m_i - 1} \pi_1 V = \frac{P_d}{\lambda}. \quad (7)$$

If these relationships hold, then the ML–PDA LR and the ML–PMHT LR are equivalent.

## III. MULTITARGET ML–PMHT

This section develops the multitarget formulation of ML–PMHT. It first discusses how it is necessary to implement ML–PDA for multiple targets in somewhat of an *ad hoc* manner. Next, it presents the natural, multitarget version of the ML–PMHT LLR along with an elegant method of optimizing it. Then, it develops a covariance estimate of the ML–PMHT state vector via the CRLB. Finally, it describes how to integrate these two concepts into a true multitarget tracking framework for ML–PMHT.

### A. ML–PDA Multitarget LLR

It is difficult to extend ML–PDA to multiple targets [10]. While it is technically possible to write the multitarget ML–PDA LLR, to take into account all the joint association events the number of terms increases rapidly with the number of targets, and the expression becomes practically intractable for any more than just a few targets. As a result, to handle multiple targets, the ML–PDA tracking framework treats scenarios with more than one target as a sequence of single-target problems. For a batch of data, it optimizes the single-target LLR (1), and if this value exceeds a certain threshold, a target is declared. (This threshold is determined by fitting the local maxima formed by the clutter to an extreme value distribution [11].) Next, the measurement that has the highest association probability with that solution is excised from each scan, and the sequence is repeated for the next target. This method is not elegant, but it has been shown to work reasonably well for multitarget scenarios in [12] and [13].

### B. ML–PMHT Multitarget LLR

In contrast to ML–PDA, the ML–PMHT LLR is very easily extended to a multitarget framework. For $K$ targets with state vectors $\mathbf{x}_1, \ldots, \mathbf{x}_K$, the multitarget LLR is expressed as

$$\Lambda'(\mathbf{x}, Z) = \sum_{i=1}^{N_w} \sum_{j=1}^{m_i} \ln \left\{ \pi_0 + V \sum_{k=1}^{K} \pi_k p[\mathbf{z}_j(i) \mid \mathbf{x}_k] \rho_{jk}(i) \right\} \quad (8)$$

where $\pi_k$ is the probability that a given measurement is from the $k$th target and $\sum_{k=0}^{K} \pi_k = 1$.

A benefit to using the ML–PMHT LLR formulation is that it is possible to optimize it with a closed-form expression using expectation–maximization (EM) [14]. The process of optimizing the ML–PMHT LLR for the single-target case was shown in [1]; here, we extend the optimization to the multitarget case. The multitarget version of the cost function is

$$J(\mathbf{x}, Z) = \sum_{k=1}^{K} \sum_{i=1}^{N_w} \sum_{j=1}^{m_i} \left\{ [\mathbf{z}_j(i) - \mathbf{H}_k \mathbf{x}]^T \mathbf{R}_{ij}^{-1} [\mathbf{z}_j(i) - \mathbf{H}_k \mathbf{x}] + \ln(|2\pi \mathbf{R}_{ij}|) \right\} w_{jk}(i). \quad (9)$$

To briefly review, in the single-target case, the ML–PMHT LLR can be optimized with EM as long as there exists a linear relationship between the predicted measurement $\bar{\mathbf{z}}$ and the state $\mathbf{x}$, where

$$\mathbf{x} = [x_0 \; \dot{x} \; y_0 \; \dot{y}]^T \quad (10)$$

and the measurement matrix is

$$\mathbf{H} = \begin{bmatrix} 1 & t & 0 & 0 \\ 0 & 0 & 1 & t \end{bmatrix}. \quad (11)$$

(For this to work, the measurements must obviously be in Cartesian space. While multistatic measurements and covariances are typically in bearing/time-delay space, they can be converted to Cartesian space following the work of [15] and [16].)

In the multitarget case for ML–PMHT, the state vector $\mathbf{x}$ becomes a $4K \times 1$ vector. [It is the vector in (10) stacked $K$ times.] The association probability $w_{jk}(i)$ is

$$w_{jk}(i) = \frac{\pi_k p[\mathbf{z}_j(i) \mid \mathbf{x_k}] \rho_{jk}(i)}{\pi_0 / V + \sum_{l=1}^{K} \pi_l p[\mathbf{z}_j(i) \mid \mathbf{x}_l] \rho_{jl}(i)}. \quad (12)$$

Matrix $\mathbf{H}_k$ in (9) is

$$\mathbf{H}_k = \begin{bmatrix} \underbrace{\begin{matrix} \cdots & 0 & \cdots \\ \cdots & 0 & \cdots \\ \cdots & 0 & \cdots \\ \cdots & 0 & \cdots \end{matrix}}_{4(k-1) \text{ columns}} & \mathbf{H} & \underbrace{\begin{matrix} \cdots & 0 & \cdots \\ \cdots & 0 & \cdots \\ \cdots & 0 & \cdots \\ \cdots & 0 & \cdots \end{matrix}}_{4(K-k) \text{ columns}} \end{bmatrix}. \quad (13)$$

This matrix is for the $k$th target; the (inner) $\mathbf{H}$ is that given by (11). There are $4(k-1)$ vectors of zeros to the left of $\mathbf{H}$ and $4(K-k)$ vectors of zeros to the right of $\mathbf{H}$. With these definitions, the solution to the problem is a simple vector quadratic minimization, and is expressed as

$$\mathbf{x} = \left[ \sum_{k=1}^{K} \sum_{i=1}^{N_w} \sum_{j=1}^{m_i} w_{jk}(i) \mathbf{H}_k^T \mathbf{R}_{ij}^{-1} \mathbf{H}_k \right]^{-1} \times \sum_{k=1}^{K} \sum_{i=1}^{N_w} \sum_{j=1}^{m_i} w_{jk}(i) \mathbf{H}_k^T \mathbf{R}_{ij}^{-1} \mathbf{z}_j(i). \quad (14)$$

### C. ML–PMHT CRLB

Here, the CRLB for ML–PMHT is developed, and it is shown that the ML–PMHT estimator is statistically efficient, i.e., the errors on the tracker state estimates meet the CRLB. Additionally, it turns out that the CRLB for ML–PMHT can be computed in real time; previous work on ML–PDA [8] showed that the CRLB for ML–PDA must be computed offline with Monte Carlo integration. Put together, these factors lead to a valuable result, namely, the CRLB calculation can be easily incorporated into the ML–PMHT tracking framework to represent the tracker error.

The full derivation for the ML–PMHT Fisher information matrix (FIM) and CRLB is presented in the Appendix; we summarize the key points to show why it is possible to compute the CRLB in real time for ML–PMHT (again, as opposed to ML–PDA). First, consider a window of ML–PMHT data with $N_w$ scans. Since all measurements from scan to scan are assumed to be independent, the FIM $\mathbf{J}$ of the total window can just be written as the sum of the individual scans

$$\mathbf{J} = \sum_{i=1}^{N_w} \mathbf{J}_i. \quad (15)$$

After some work (again, presented in the Appendix), a key point arises. The expression for $\mathbf{J}_i$ is given as (16) at the bottom of the page. Here, $\phi$ is the state-to-measurement conversion, and $\mathbf{D}_\phi$ is the Jacobian of $\phi$. Next, $\mathbf{R}_j$ is the measurement covariance for the $j$th measurement in the scan. [Since (16) only deals with a single scan, the $i$-subscripts are dropped.] Finally, $d\mathbf{Z} = \prod_{l=1}^{m_i} d\mathbf{z}_l$ and $d\mathbf{a} = \prod_{l=1}^{m_i} da_l$. All $p(\mathbf{z}_l \,|\, \mathbf{x})$ terms for $l \neq j$ are separable and integrate to one, and the term for $l = j$ is canceled by one of the terms in the denominator. All that is left within the integral is the single measurement $\mathbf{z}_j$. This reduces the calculation to a dimension equal to the number of dimensions in the measurement, plus one for the amplitude. This makes it possible for ML–PMHT to compute the FIM (and, thus, the CRLB) in real time as part of the tracking framework. In contrast, for the ML–PDA FIM calculation, a summation over all $m_i$ measurements in the scan remains in the denominator [8]. Thus, the dimensionality of the calculation in this case is approximately $m_i$ times the number of measurement dimensions. As a result, the ML–PDA FIM must be calculated offline with Monte Carlo integration.

After some work, the final expression for the ML–PMHT FIM is

$$\mathbf{J}_i = \mathbf{D}_\phi^T \sum_{j=1}^{m_i} \mathbf{G}_j^T \times \int_V \frac{\dfrac{[\pi_1 p_1^\tau(a_j)]^2}{|2\pi\mathbf{R}_j|} e^{-\xi_j^T \xi_j} \xi_j \xi_j^T}{\dfrac{\pi_0 p_0^\tau(a_j)}{V} + \dfrac{\pi_1 p_1^\tau(a_j)}{\sqrt{|2\pi\mathbf{R}_j|}} e^{-\frac{1}{2}\xi_j^T \xi_j}} d\xi_j da_j \frac{\mathbf{G}_j}{|\mathbf{G}_j|} \mathbf{D}_\phi \quad (17)$$

where $\xi_j$ is a dummy variable of integration with dimensionality equal to the measurement dimensionality, and $\mathbf{G}_j$ is the Cholesky decomposition of $\mathbf{R}_j^{-1}$. This is inserted into (15), and then the CRLB is just taken as the inverse of the FIM.

At this point, the efficiency of the ML–PMHT estimator was checked by using this FIM to calculate the normalized estimation error squared (NEES), which is given by

$$\text{NEES} = (\hat{\mathbf{x}} - \mathbf{x}_{\text{truth}})^T \mathbf{J} (\hat{\mathbf{x}} - \mathbf{x}_{\text{truth}}). \quad (18)$$

Scenario 1 (described below) was run 500 times, and for each run, the median NEES value for a valid track was saved (there were multiple NEES values for a run due to the sliding batch nature of the tracker implementation). The average of these NEES values came out to 4.15; the expected NEES for a four-parameter state vector is 4, with a 95% confidence interval of [3.81, 4.19]. The average NEES value falls within this confidence interval, so the conclusion is that ML–PMHT is a statistically efficient estimator. A subset of 20 of these runs is shown in Fig. 1 with the 95% position uncertainty ellipses at the beginning and end of the runs plotted. As expected, almost all of the solution points fall within their respective ellipses. The CRLB for ML–PMHT (as opposed to ML–PDA) can be computed in real time and then be used to accurately represent the error on the state estimate from the tracker.

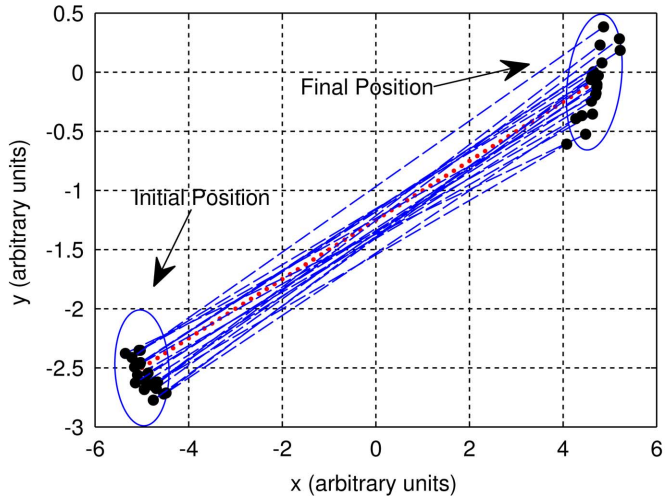


Fig. 1. CRLB 95% uncertainty regions for initial and final positions.

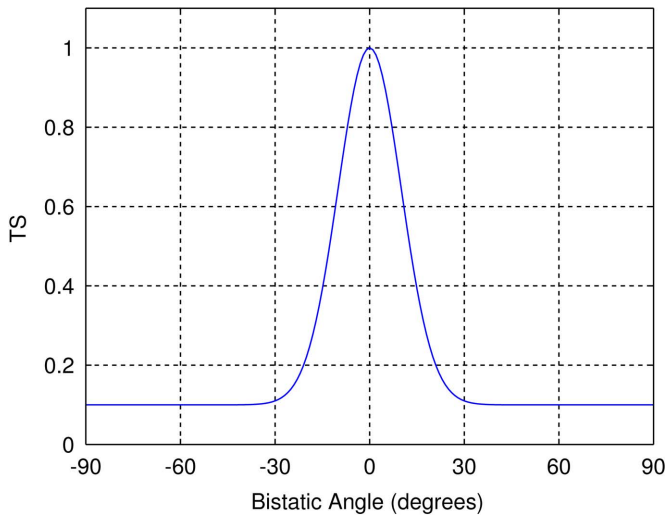


Fig. 2. TS as a function of bistatic angle.

### D. ML–PMHT Multitarget Implementation

The ML–PMHT multitarget tracking framework was implemented in the track update sequence by testing all existing tracks for "closeness." Any tracks that were determined to be close were grouped together and optimized with the multitarget ML–PMHT LLR formulation in (8). Grouping tracks involved estimating the state covariance of all existing tracks. We let the CRLB developed above represent the state covariance $\mathbf{C}_n$ for the $n$th target, and with this, a $\chi^2$ test statistic [17] is evaluated

$$\mathbf{J}_i = \sum_{j=1}^{m_i} \int_V \mathbf{D}_\phi^T \frac{\dfrac{[\pi_1 p_1^\tau(a_j)]^2}{|2\pi\mathbf{R}_j|} e^{-(\mathbf{z}_j - \boldsymbol{\phi})^T \mathbf{R}_j^{-1} (\mathbf{z}_j - \boldsymbol{\phi})} \mathbf{R}_j^{-1} (\mathbf{z}_j - \boldsymbol{\phi})(\mathbf{z}_j - \boldsymbol{\phi})^T \mathbf{R}_j^{-1}}{p(\mathbf{z}_j \,|\, \mathbf{x})^2} \mathbf{D}_\phi \prod_{l=1}^{m_i} p(\mathbf{z}_l \,|\, \mathbf{x}) d\mathbf{Z} d\mathbf{a}. \quad (16)$$

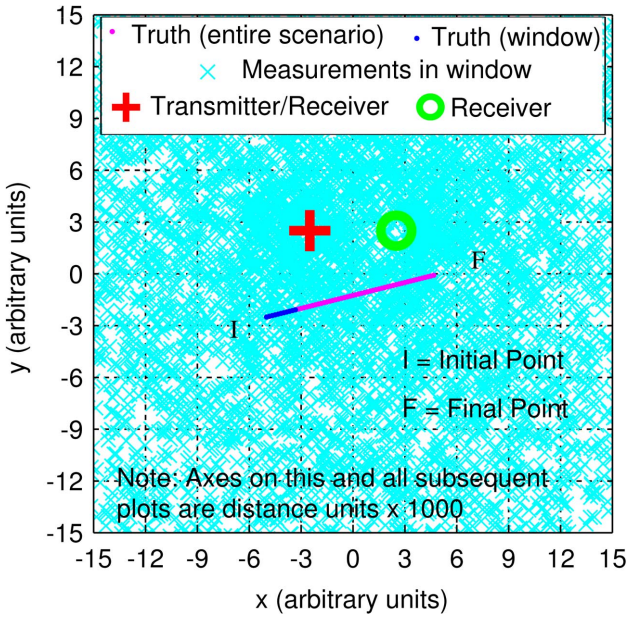


Fig. 3. All measurements from a window of data (11 update periods) for scenario 1 with Rayleigh clutter.

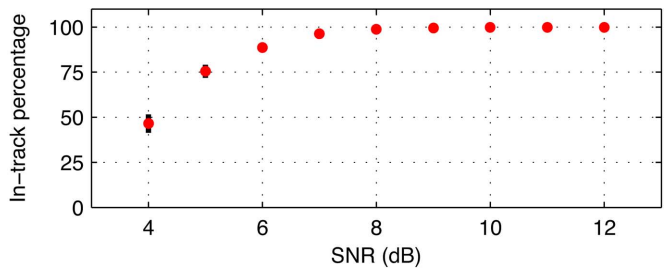


Fig. 4. ML–PDA in-track percentage versus expected target SNR for scenario 1 with Rayleigh clutter.

for closeness between all possible pairs of active tracks in the form

$$S_{mn} = \Delta\mathbf{x}_{mn}^T(\mathbf{C}_m + \mathbf{C}_n)^{-1}\Delta\mathbf{x}_{mn} \tag{19}$$

where $\Delta\mathbf{x}_{mn} = \mathbf{x}_m - \mathbf{x}_n$ (the difference of the state vector estimates between the $m$th and $n$th track). All track pairs with a statistic $S_{mn}$ less than a given threshold are grouped together. More than two tracks could belong to a group by the use of this association; to join a group, an ungrouped track only had to meet the test described in (19) with one of the tracks already in the group.

## IV. Simulator and Simulations

This section describes in detail the simulator that was used to create the data for the Monte Carlo runs and then presents the five scenarios used to compare ML–PDA and ML–PMHT.

### A. Simulator Description

The Monte Carlo runs were performed using a multistatic simulator developed at the University of Connecticut (Storrs, CT, USA) [18]. The simulator takes as input the source and receiver positions (all were stationary in this work), as well as the number of targets and target geometries; this information is all

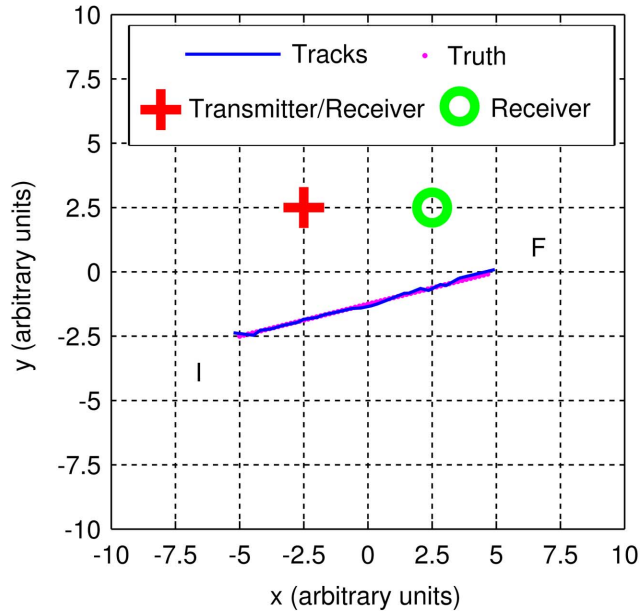


Fig. 5. Scenario 1 (baseline) and ML–PDA sample estimated track in one run.

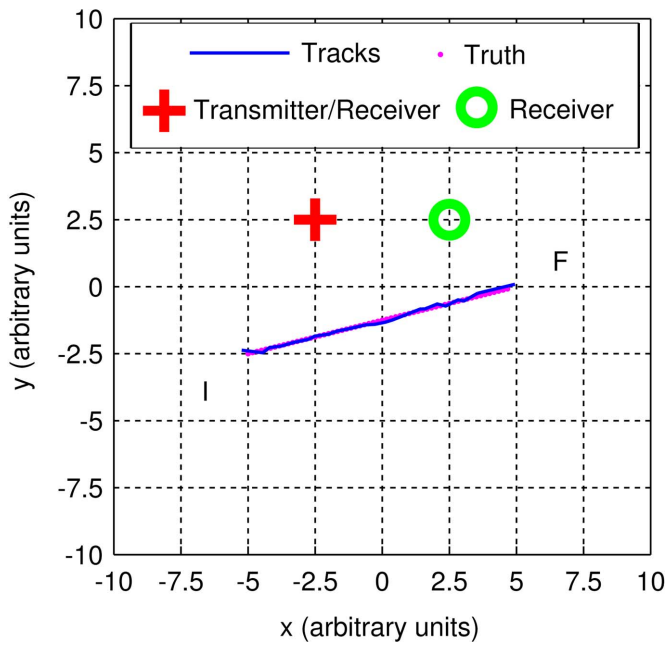


Fig. 6. Scenario 1 (baseline) and ML–PMHT sample estimated track in one run.

shown in Figs. 2–14. Each transmitter generates a ping every 60 s, and for each source–receiver pair (i.e., a scan), the simulator calculates the number, position, amplitude, and Doppler of the clutter points, and the existence, position, amplitude, and Doppler of the target returns. (All pings in the simulation were treated as being Doppler sensitive.)

*1) Clutter Measurement Generation:* First, to generate the clutter points, the simulator calculates the number of resolution cells that should be present. The number of resolution cells is treated as a function of $\sigma_{\text{az}}$ and $\sigma_t$, the azimuthal and time-delay errors simulated for the system. The number of azimuthal resolution cells is approximated as

$$N_{\text{az}} = \frac{360}{\text{BW}} \tag{20}$$

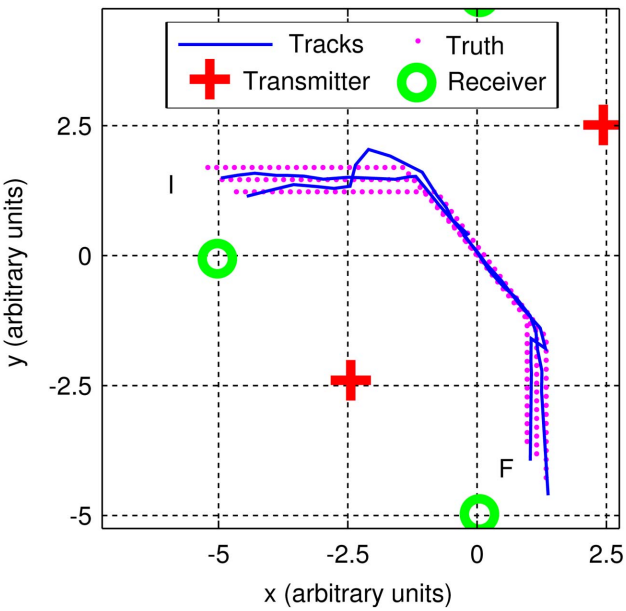


Fig. 7. Scenario 2 (three close targets with similar dynamics) and ML–PDA sample estimated tracks in one run.

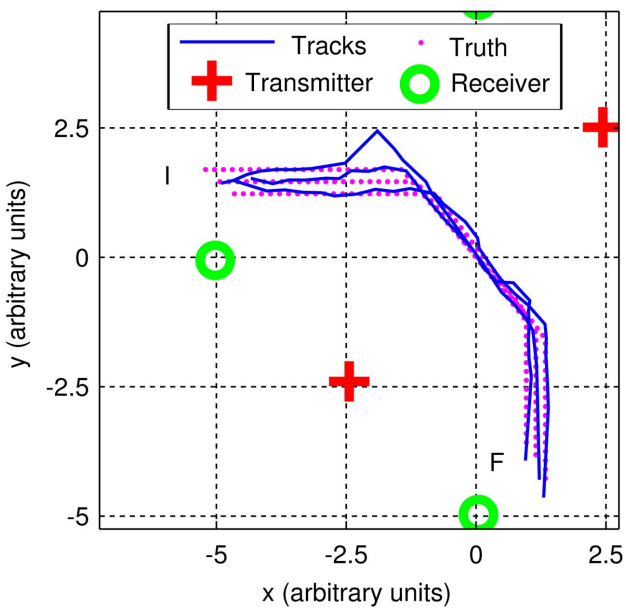


Fig. 8. Scenario 2 (three close targets with similar dynamics) and ML–PMHT sample estimated tracks in one run.

where BW is the beamwidth of a receive beam in degrees. The simulator does not actually form any beams; to get a value of BW, it makes the assumption that the clutter is distributed uniformly in a beam, which gives $\sigma_{\text{az}} = \text{BW}/\sqrt{12}$. This, in turn, leads to a final expression for the number of azimuthal resolution cells

$$N_{\text{az}} = \left\lfloor \frac{360}{\sqrt{12}\sigma_{\text{az}}} \right\rfloor. \tag{21}$$

(Here, "$\lfloor\,\rfloor$" signifies round toward the next lower integer.) For all of the scenarios in this work, the azimuthal uncertainty was $\sigma_{\text{az}} = 5^\circ$ (this and all other simulated error values are provided

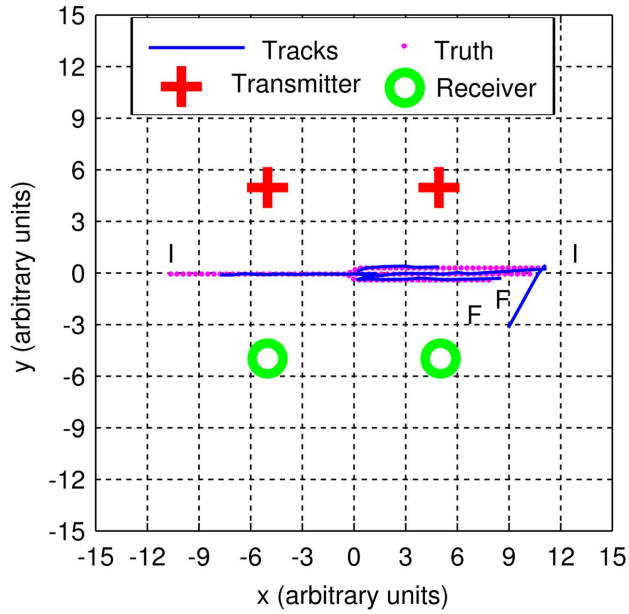


Fig. 9. Scenario 3 (two close targets with different dynamics) and ML–PDA sample estimated tracks in one run.

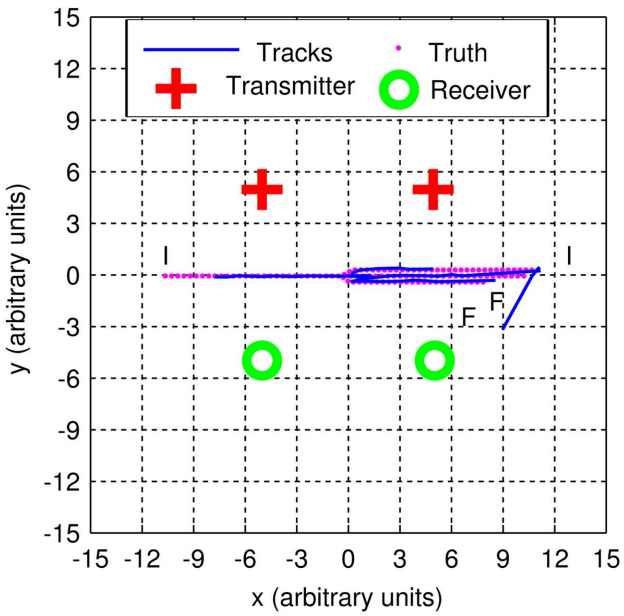


Fig. 10. Scenario 3 (two close targets with different dynamics) and ML–PMHT sample estimated tracks in one run.

together in Table II). Using similar logic, the number of delay (range) resolution cells is calculated with the expression

$$N_{\text{delay}} = \left\lfloor \frac{t_{\text{ipt}}/2 - t_{\text{bt}}}{\sqrt{12}\sigma_t} \right\rfloor. \tag{22}$$

Here, $t_{\text{ipt}}$ is inter-ping time (60 s throughout this work), and $t_{\text{bt}}$ is blanking time. This value is determined by the amount of time sound takes to travel from the source to the receiver. Finally, the number of resolution cells is simply given by the product of $N_{\text{az}}$ and $N_{\text{delay}}$.

With the number of resolution cells determined, the clutter is generated. For each scan, to simulate the amplitudes of the clutter distribution, a number of random variables equal to the

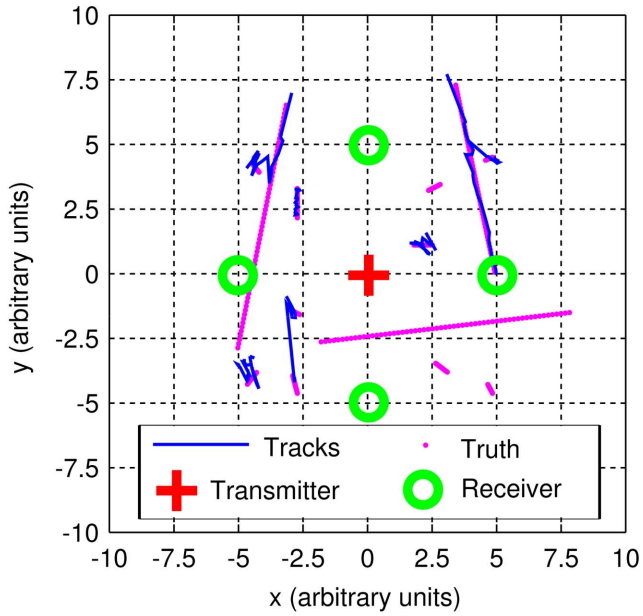


Fig. 11. Scenario 4 (large number of targets) and ML–PDA sample estimated tracks in one run.

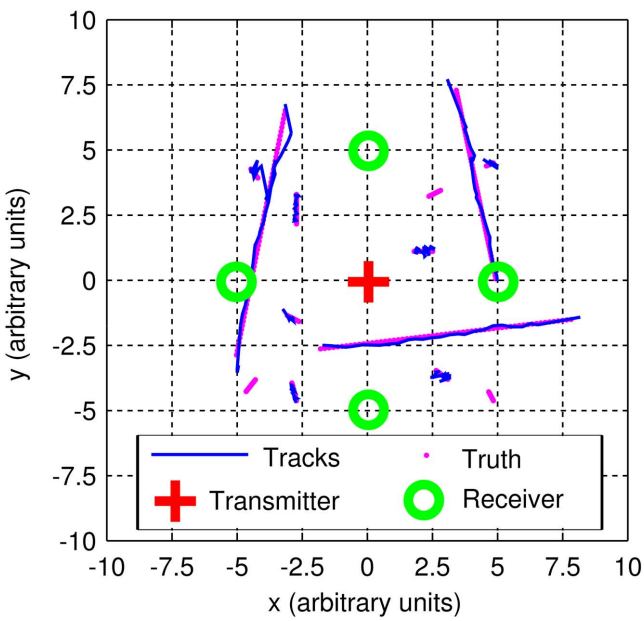


Fig. 12. Scenario 4 (large number of targets) and ML–PMHT sample estimated tracks in one run.

number of resolution cells is generated. For the first five scenarios, the amplitude of this clutter distribution was given a $K$-distribution. The probability distribution for the intensity of the clutter (i.e., the amplitude squared) is given by [19], [20]

$$f(p) = \frac{2}{\lambda \Gamma(\alpha_k)} \left( \frac{p}{\lambda_k} \right)^{(\alpha_k - 1)/2} K_{\alpha_k - 1} \left( 2\sqrt{\frac{p}{\lambda_k}} \right) \qquad (23)$$

where $p > 0$. Here, $\alpha_k$ is a $K$-distribution parameter ($\alpha_k = 0.5$ throughout this work), $\lambda_k = 1/\alpha_k$, and $K_\nu$ is the Basset function (a modified Bessel function of the second kind) [21].

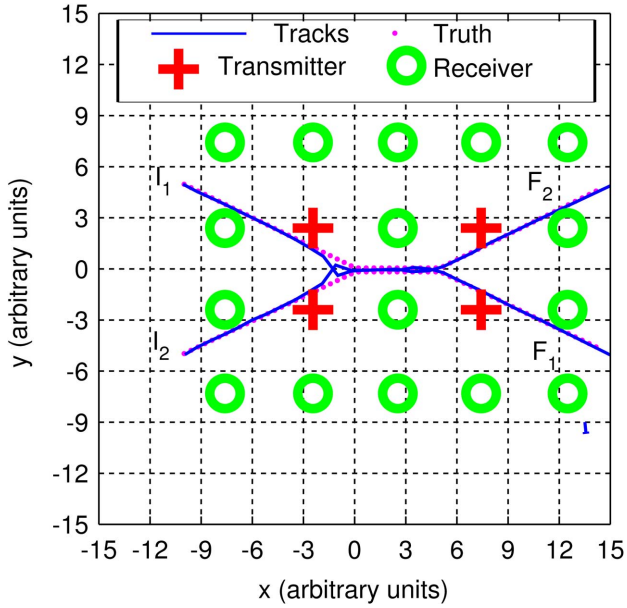


Fig. 13. Scenario 5 (switching targets) and ML–PDA sample estimated tracks in one run (switch occurred).

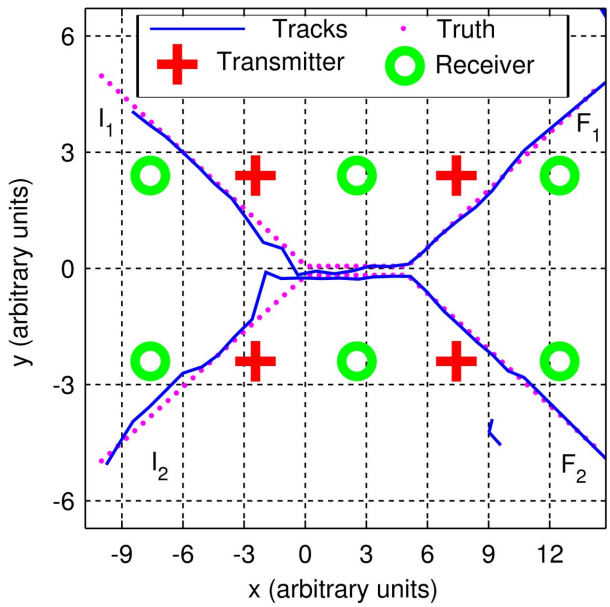


Fig. 14. Scenario 5 (switching targets) and ML–PMHT sample estimated tracks in one run, zoomed-in view (no switching occurred).

Random variables from the distribution shown in (23) are generated via [20]

$$K = -\Gamma \lambda_k \ln U \qquad (24)$$

where $U$ is a realization of a uniform random variable over the interval [0, 1], and $\Gamma$ is a realization of a Gamma random variable with shape parameter $\alpha = \alpha_k$ and a scale parameter $\beta = 1$.

For clarity, we note that the simulator generates "amplitude" LLRs by operating in intensity (power) space, i.e., amplitude squared. Such LLRs can be computed in either amplitude or intensity space; as long as the clutter and target distributions are consistent in terms of the units used, the LLR values will come out the same.

TABLE I
DETECTOR THRESHOLDS

| Scenario | Threshold (dB) |
|---|---|
| 1-4 | 12 |
| 5 | 14 |
| 1 (Rayleigh clutter) | 2 |

TABLE II
SIMULATION ERRORS

| $\sigma_{az}$ (deg) | $\sigma_t$ (s) | $\sigma_{\text{Doppler target}}$ (units/s) | $\sigma_{\text{Doppler clutter}}$ (units/s) |
|---|---|---|---|
| 5 | 0.1 | 0.5 | 2 |

For the final scenario, the clutter amplitude was given a Rayleigh distribution following the work in [8], so the intensity of the clutter followed an exponential distribution. The clutter intensity in this case was simulated by generating exponential random variables with a mean of one.

Once the clutter realizations were created, they were thresholded. Amplitudes above the threshold were kept; amplitudes below the threshold were discarded. This threshold is what determined the clutter density; it was set so that the clutter density for the average source–receiver pair in the first five scenarios was approximately $1 \times 10^{-9}$ clutter points per unit volume; for the final scenario with Rayleigh clutter, the threshold was set so that the clutter density was approximately $1 \times 10^{-7}$ clutter points per unit volume. The amplitude detector threshold values (in decibels) are given in Table I. After this, all remaining clutter points were given delay time and azimuth measurements by sampling from a uniform distribution over the proper time and azimuthal interval. Finally, all generated clutter measurements were given a Doppler component. The assumption was made that the clutter would be stationary; thus, the Doppler measurements (simulated in the form of bistatic range rate) were generated from a zero-mean Gaussian with $\sigma_{\text{Doppler clutter}} = 2$ units/s.

*2) Target Measurement Generation:* For every update, the target and source positions were updated according to user-defined course and speed segments. The expected target-reflected power at the receiver was then calculated (in decibels) with

$$P_{\text{rec}} = \text{SL}_{1000} - 10\log_{10}(R_{\text{ST}}/1000) - 10\log_{10}(R_{\text{TR}}/1000) + 10\log_{10}(\text{TS}). \quad (25)$$

Here, $\text{SL}_{1000}$ is the signal power 1000 distance units from the target (all distance units in this simulation are arbitrary), $R_{\text{TS}}$ is the distance from the target to the source, $R_{\text{TR}}$ is the distance from the target to the receiver, and TS is the bistatic target strength. This bistatic target strength is a function of the bistatic angle, defined in [16]. For this simulation, the value of TS as a function of bistatic angle is shown in Fig. 2. [Note that with (25) we are not trying to implement directly the physical effects described by the active sonar equation. Instead, we are trying to implement a target amplitude model that has a known dependency on range and bistatic aspect and is given a reference level at a convenient, fixed distance from the target.] Finally, the "measured" target amplitude was simulated having a Rayleigh distribution, with expected amplitude given by (25). The measured target amplitude random variable is simulated (in decibel space) with

$$P_{\text{tgt meas}} = P_{\text{rec}} + 10\log_{10}(-\ln(U)). \quad (26)$$

Here, $U$ is a uniform random variable in the interval [0, 1]. [It is easy to show that if $P_{\text{tgt meas}}$ is converted to intensity space, (26) produces a realization of an exponential random variable with mean $10^{P_{\text{rec}}/10}$.] If target return power $P_{\text{tgt meas}}$ is greater than the respective threshold for its scenario shown in Table I, then the (target-originated) detect is kept in the data for processing. The values of $\text{SL}_{1000}$ are shown in Table III.

TABLE III
$\text{SL}_{1000}$ VALUES AS A FUNCTION OF SCENARIO

| Scenario | 1 | 2 | 3 | 4 | 5 | 1 (Rayleigh clut) |
|---|---|---|---|---|---|---|
| $SL_{1000}$ (dB) | 35 | 40 | 45 | 40 | 45 | 27 |

Finally, the time, azimuth, and bistatic range-rate values for all target measurements are calculated and then corrupted with noise. Based on the (deterministic) positions of the source, the receiver, and the target, delay time $t$ and azimuth $\theta$ (from the receiver to the target) are determined via simple geometry. These values of $t$ and $\theta$ are corrupted by adding realizations of zero-mean Gaussian random variables with variances $\sigma_t^2$ and $\sigma_{\text{az}}^2$ to them, respectively. The bistatic range rate is calculated following the work of [16], and then this is corrupted with a zero-mean Gaussian random variable with variance $\sigma^2_{\text{Doppler target}}$.

*3) ML–PMHT Target Measurement Generation Model:* The majority of simulations in this work followed the ML–PDA target measurement generation model, that is, at most one measurement originated from the target in a given scan (this is implicit in the simulator discussion above). However, the simulator was also set up with the ability to generate multiple target measurements per scan, i.e., the ML–PMHT target measurement generation model. This was done by assuming $P_d$ is known, and then writing the probability mass function (PMF) describing the number of target measurements $f(x, \lambda_t)$ as

$$f(x,\lambda_t) = \begin{cases} 1 - P_d, & x = 0 \\ \dfrac{P_d}{1 - e^{-\lambda_t}} \dfrac{e^{-\lambda_t}\lambda_t^x}{x!}, & x > 0. \end{cases} \quad (27)$$

Let $N = E[x]$, the expected number of measurements from the PMF. Calculating the expected value of (27) produces the following relationship:

$$N(1 - e^{-\lambda_t}) = P_d\lambda_t. \quad (28)$$

Again, the value of $P_d$ is assumed known (in this work, it was $P_d \approx 0.8$). Then, (28) can be numerically solved for $\lambda_t$ as a function of $N$. Results of this are presented in Table IV. To produce, on average, $N$ target-generated measurements per scan, the simulator first determines if the contact power is greater than the threshold, as described above. If the target power does exceed the threshold, the simulator draws a realization from a truncated Poisson PMF with parameter $\lambda_t$, and creates this many

TABLE IV
VALUES FOR $\lambda_t$ AS A FUNCTION OF THE EXPECTED NUMBER OF TARGET MEASUREMENTS

| N | $\lambda_t$ |
|---|---|
| 1 | 0.4642 |
| 2 | 2.2316 |
| 3 | 3.5628 |

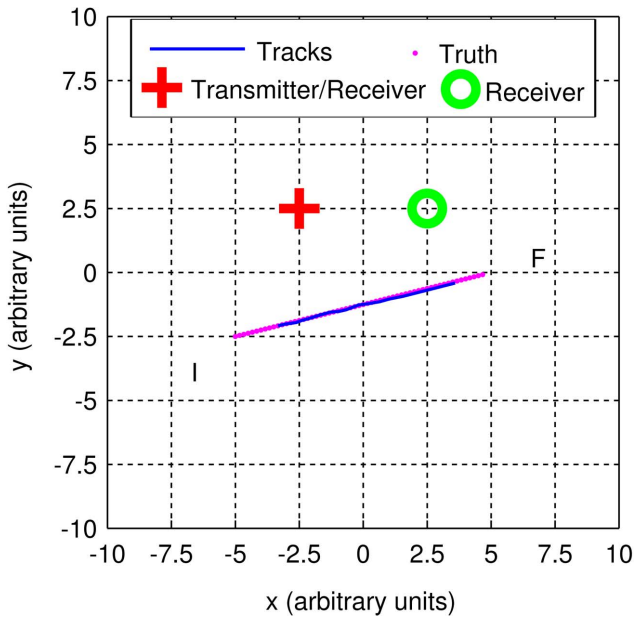


Fig. 15. Scenario 1 (baseline) and ML–PDA sample estimated track with high density Rayleigh clutter in one run.

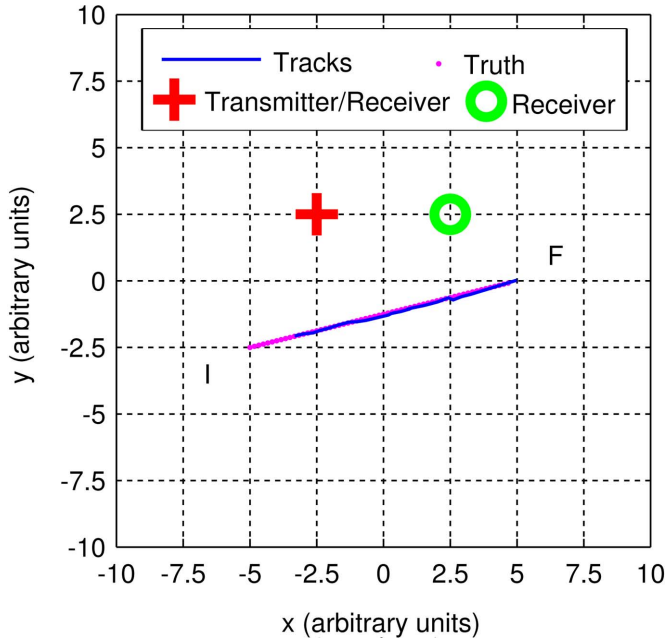


Fig. 16. Scenario 1 (baseline) and ML–PMHT sample estimated track with high density Rayleigh clutter in one run.

TABLE V
ML–PMHT AND ML–PDA PARAMETERS

| | ML-PMHT | | | ML-PDA | |
|---|---|---|---|---|---|
| | $\pi_0$ | $\pi_1$ | $V$ | $\lambda$ | $P_d$ |
| Scen. 1-4 | 0.95 | 0.05 | $1.26 \times 10^9$ | $3.7 \times 10^{-9}$ | 0.8 |
| Scen. 5 | 0.95 | 0.05 | $1.26 \times 10^9$ | $6.7 \times 10^{-10}$ | 0.8 |
| Scen. 1 Rayleigh clut. | 0.95 | 0.05 | $1.26 \times 10^9$ | $2.0 \times 10^{-7}$ | 0.8 |

(target) measurements, again, in the same manner as described above.

### B. Scenarios

Five benchmark multistatic sonar scenarios, initially developed in [1], were used to measure performance differences between ML–PDA and ML–PMHT with Monte Carlo testing. Each scenario was designed so target detections from a single source–receiver pair would be present approximately 80% of the time in a given scan, and clutter was set at a level that made the problem as difficult as possible while not slowing down run times to the point of precluding Monte Carlo testing. The various scenario parameters are listed in Table V (these parameters, used in the simulation, were also matched in the actual ML–PDA and ML–PMHT tracking code). All scenarios are shown (with example results overlaid) in Figs. 3–16. Each of the scenario target geometries was designed for a specific test purpose. Then, given these target geometries, the sensors were laid out in an effort to obtain an average target $P_d$ of 0.8 in any given scan. (The one exception to this occurs in scenario 5 and is described below.) The pinging strategy was purposely kept very simple; every transmitter had a (simulated) ping every 60 s. The scenarios are described as follows.

*1) Baseline Scenario:* Scenario 1, shown in Fig. 3, featured a single target moving in a straight line past a source and a receiver (the source was a receiver as well). This was intentionally created as one of the simplest possible multistatic scenarios. As is shown in Section II, ML–PDA and ML–PMHT should converge to the same value under benign conditions, and this scenario is a case where this convergence should hold true.

*2) Close Targets With Similar Dynamics Scenario:* Scenario 2, shown in Fig. 7, features three targets very close together (they have a separation of only 500 distance units) with similar motion dynamics, i.e., they are maneuvering in a coordinated fashion. This makes it very difficult for any algorithm to distinguish and separate targets from each other. This will test if the multitarget implementation for ML–PMHT is an improvement over the sequential single-target implementation of ML–PDA.

*3) Close Targets With Different Dynamics Scenario:* Scenario 3, shown in Fig. 9, has two targets closing each other from the east and west. During the middle of the scenario, the targets will be in close proximity to each other, but they will have different motion dynamics; at the time they are close, they will be moving in opposite directions. Again, this is designed to test each tracker's ability to follow close targets, but under slightly easier circumstances than in scenario 2.

*4) Large Number of Targets Scenario:* Scenario 4, shown in Fig. 11, features ten very low-speed targets and three relatively high-speed targets. This scenario was designed simply to measure each algorithm's ability to track a (relatively) large number

of targets, especially when some of those targets are low speed and exhibit very low Doppler.

*5) Switching Targets Scenario:* Scenario 5, shown in Fig. 13, is similar to scenario 2 in that it has two targets in close proximity to each other with the same motion dynamics. However, the targets start out with a distinct separation and then close on each other. At the point where the targets are about to intersect, they turn and parallel each other. Such motion, with the targets approaching each other (where their projected motion has them crossing), will make it very difficult for any trackers continually to associate measurements with the correct targets and not to switch targets. Additionally, this scenario was given a relatively high number of receivers (16), and due to the geometry, most source receiver pairs will have a $P_d$ of much less than 0.8; many scans will just have clutter measurements. This is effectively raising the clutter levels that the tracker will see.

These scenarios were all run using the ML–PDA target measurement generation model, where the target produced at most one measurement per scan. Scenario 1 was then rerun with the ML–PMHT target measurement generation model, where more than one measurement was allowed to be generated by the target in a scan. This scenario was run three times following the target measurement generation method described in Section IV-A3; the expected number of target measurements per scan was one, two, and three, respectively. (Note the case of one target measurement per scan is not the ML–PDA target measurement generation model. In this case, on average, the target generates one measurement per scan, but there could be more. In the ML–PDA case, the number of target measurements per scan is strictly limited to zero or one.)

At the conclusion of each scenario, the following metrics were evaluated: target in-track percentage, target position root mean square error (RMSE), track fragmentation, target duplicate tracks, overall number of false tracks, and average false track length. These metrics were calculated in the following manner. First, tracks were associated with either a target or clutter. A track was associated with a target if its NEES value was less than 10 for scenarios with a single target or widely spaced targets. (In most situations, as is discussed above, the estimators were statistically efficient.) For scenarios with closely spaced maneuvering targets, the CRLB tended to underestimate the covariance (the estimators were not statistically efficient here), so the NEES was not as reliable in associating tracks to targets. Instead, in these cases, a track was associated with a target if the average distance between track points and their associated truth points was less than 2000 distance units. If a track could be associated with more than one target, then it was assigned to whichever one was closest. Target in-track percentage was then calculated by taking the ratio of target truth points that had a track associated with them to the total number of target truth points. Target duplicate tracks were calculated by simply counting the number of targets that had more than one track associated with them at a given time. The RMSE was then calculated for all tracks that were associated with targets. This RMSE value was only calculated for a target on points where there were both an ML–PDA track and an ML–PMHT track associated with it. This was done so as not to penalize one algorithm that was barely holding track (with a resultant poor RMSE) while the other algorithm was not tracking at all. Track fragmentation was determined by counting the number of breaks in track for a target. Finally, the overall number of false tracks was simply the number of tracks that were not associated with a target, and the mean false track length was the average number of updates for which a false track was active.

For each of the Monte Carlo sets, 200 runs were performed. Additionally, the confidence intervals shown in Tables VI–XII are at the 95% level.

TABLE VI
IN-TRACK PERCENTAGE RESULTS

| | ML-PDA | | ML-PMHT | |
|---|---|---|---|---|
| | mean | confidence interval (95 percent) | mean | confidence interval (95 percent) |
| Scenario 1 | 95.4 | [94.8, 96.0] | 95.4 | [94.8, 96.0] |
| Scenario 2 Tgt 1 | 12.0 | [7.5, 16.6] | 83.9 | [79.6, 88.2] |
| Scenario 2 Tgt 2 | 100 | [100, 100] | 87.5 | [83.7, 91.3] |
| Scenario 2 Tgt 3 | 1.9 | [0.0, 3.8] | 53.1 | [46.5, 59.7] |
| Scenario 3 Tgt 1 | 82.7 | [81.0, 84.4] | 82.5 | [81.0, 84.0] |
| Scenario 3 Tgt 2 | 76.1 | [74.2, 78.0] | 77.1 | [75.4, 78.9] |
| Scenario 4 Tgt 7 | 61.0 | [54.7, 67.4] | 91.1 | [89.3, 92.8] |
| Scenario 4 Tgt 12 | 52.5 | [45.9, 59.2] | 96.3 | [95.9, 96.8] |
| Scenario 5 Tgt 1 | 45.3 | [38.4, 52.2] | 63.1 | [56.5, 69.7] |
| Scenario 5 Tgt 2 | 48.4 | [41.5, 55.4] | 63.7 | [57.1, 70.2] |
| Scenario 1 Rayleigh | 66.9 | [66.0, 67.8] | 67.4 | [66.6, 68.3] |

TABLE VII
RMSE RESULTS

| | ML-PDA | | ML-PMHT | |
|---|---|---|---|---|
| | mean | confidence interval (95 percent) | mean | confidence interval (95 percent) |
| Scenario 1 | 255.8 | [229.7, 281.9] | 259.2 | [233.0, 285.4] |
| Scenario 2 Tgt 1 | 313.4 | [257.7, 369.0] | 281.4 | [267.7, 295.1] |
| Scenario 2 Tgt 2 | 247.0 | [241.9, 252.1] | 280.5 | [269.6, 291.5] |
| Scenario 2 Tgt 3 | 228.4 | [186.7, 270.0] | 212.4 | [202.3, 222.4] |
| Scenario 3 Tgt 1 | 1227.3 | [1190.2, 1264.3] | 946.6 | [904.4, 988.8] |
| Scenario 3 Tgt 2 | 799.7 | [754.6, 844.8] | 878.0 | [836.1, 920.0] |
| Scenario 4 Tgt 7 | 391.3 | [303.8, 478.8] | 321.6 | [274.5, 377.7] |
| Scenario 4 Tgt 12 | 151.6 | [124.0, 179.2] | 213.2 | [177.2, 249.3] |
| Scenario 5 Tgt 1 | 213.1 | [173.3, 252.8] | 244.7 | [207.1, 282.2] |
| Scenario 5 Tgt 2 | 272.5 | [211.3, 333.7] | 245.6 | [215.2, 276.1] |
| Scenario 1 Rayleigh | 217.6 | [191.7, 243.6] | 182.4 | [166.4, 198.9] |

## V. ML–PDA VERSUS ML–PMHT PERFORMANCE COMPARISONS

Both ML–PDA and ML–PMHT showed themselves to be excellent trackers for targets with low amplitude returns (detailed below). First, to get an idea of the level of difficulty of the problem, all measurements from a batch are plotted in Fig. 3 for scenario 1 with the Rayleigh clutter distribution (the scenario with the highest level of clutter).

Using this same scenario, to provide a comparison to other algorithms with known performance, in-track percentage as a function of SNR was calculated. This is shown in Fig. 4 for ML–PDA (results were practically identical for ML–PMHT). For this example (in contrast to the rest of the runs), expected

TABLE VIII
FRAGMENTATION RESULTS

| | ML-PDA | | ML-PMHT | |
|---|---|---|---|---|
| | mean | confidence interval (95 percent) | mean | confidence interval (95 percent) |
| Scenario 1 | 0.01 | [0.00, 0.02] | 0.01 | [0.00, 0.02] |
| Scenario 2 Tgt 1 | 0.00 | [0.00, 0.00] | 0.00 | [0.00, 0.00] |
| Scenario 2 Tgt 2 | 0.00 | [0.00, 0.00] | 0.00 | [0.00, 0.00] |
| Scenario 2 Tgt 3 | 0.00 | [0.00, 0.00] | 0.00 | [0.00, 0.00] |
| Scenario 3 Tgt 1 | 0.03 | [0.01, 0.06] | 0.07 | [0.03, 0.10] |
| Scenario 3 Tgt 2 | 0.24 | [0.17, 0.30] | 0.06 | [0.02, 0.09] |
| Scenario 4 Tgt 7 | 0.00 | [0.00, 0.00] | 0.00 | [0.00, 0.00] |
| Scenario 4 Tgt 12 | 0.00 | [0.00, 0.00] | 0.00 | [0.00, 0.00] |
| Scenario 5 Tgt 1 | 0.15 | [0.07, 0.22] | 0.15 | [0.09, 0.22] |
| Scenario 5 Tgt 2 | 0.09 | [0.03, 0.15] | 0.15 | [0.08, 0.22] |
| Scenario 1 Rayleigh | 0.07 | [0.03, 0.10] | 0.05 | [0.02, 0.08] |

TABLE IX
DUPLICATE TRACK RESULTS

| | ML-PDA | | ML-PMHT | |
|---|---|---|---|---|
| | mean | confidence interval (95 percent) | mean | confidence interval (95 percent) |
| Scenario 1 | 0.18 | [0.13, 0.24] | 0.02 | [0.00, 0.04] |
| Scenario 2 Tgt 1 | 0.00 | [0.00, 0.00] | 0.04 | [0.01, 0.08] |
| Scenario 2 Tgt 2 | 1.07 | [1.01, 1.14] | 0.48 | [0.39, 0.57] |
| Scenario 2 Tgt 3 | 0.00 | [0.00, 0.00] | 0.00 | [0.00, 0.00] |
| Scenario 3 Tgt 1 | 0.03 | [0.01, 0.06] | 0.07 | [0.03, 0.10] |
| Scenario 3 Tgt 2 | 0.24 | [0.17, 0.30] | 0.06 | [0.02, 0.09] |
| Scenario 4 Tgt 7 | 0.00 | [0.00, 0.00] | 0.09 | [0.05, 0.13] |
| Scenario 4 Tgt 12 | 0.09 | [0.03, 0.14] | 0.18 | [0.12, 0.24] |
| Scenario 5 Tgt 1 | 0.04 | [0.00, 0.08] | 0.13 | [0.07, 0.19] |
| Scenario 5 Tgt 2 | 0.02 | [0.00, 0.05] | 0.11 | [0.05, 0.16] |
| Scenario 1 Rayleigh | 2.10 | [1.83, 2.37] | 0.00 | [0.00, 0.00] |

TABLE X
FALSE TRACK RESULTS

| | ML-PDA | | ML-PMHT | |
|---|---|---|---|---|
| | mean | confidence interval (95 percent) | mean | confidence interval (95 percent) |
| Scenario 1 | 0.07 | [0.03, 0.11] | 0.07 | [0.03, 0.11] |
| Scenario 2 | 0.06 | [0.03, 0.09] | 0.15 | [0.15, 0.21] |
| Scenario 3 | 0.34 | [0.26, 0.42] | 0.26 | [0.19, 0.33] |
| Scenario 4 | 0.17 | [0.10, 0.23] | 0.14 | [0.09, 0.20] |
| Scenario 5 | 2.13 | [1.92, 2.34] | 2.65 | [2.44, 2.86] |
| Scenario 1 Rayleigh | 0.08 | [0.02, 0.14] | 0.00 | [0.00, 0.00] |

target SNR was fixed, starting at 4 dB with a 2-dB measurement threshold. The expected target SNR was then moved up in 1-dB increments, with Monte Carlo simulations being run at each SNR level. Note that the tracker achieves a $P_d$ of approximately 50% at a target SNR of only 4 dB, and a $P_d$ of about 90% at 6 dB. By a target SNR of 8 dB, the tracker essentially has a $P_d$ of 100%. This shows that ML–PDA and ML–PMHT are effective VLO trackers.

TABLE XI
FALSE TRACK LENGTH RESULTS

| | ML-PDA | | ML-PMHT | |
|---|---|---|---|---|
| | mean | confidence interval (95 percent) | mean | confidence interval (95 percent) |
| Scenario 1 | 3.14 | [2.59, 3.69] | 3.14 | [2.56, 3.73] |
| Scenario 2 | 2.75 | [2.31, 3.19] | 3.37 | [2.81, 3.92] |
| Scenario 3 | 4.34 | [3.18, 5.50] | 4.42 | [3.32, 5.62] |
| Scenario 4 | 6.75 | [3.72, 9.78] | 11.67 | [7.81, 15.52] |
| Scenario 5 | 8.98 | [7.81, 10.16] | 8.55 | [7.59, 9.51] |
| Scenario 1 Rayleigh | 3.58 | [2.39, 4.78] | 0.00 | [0.00, 0.00] |

TABLE XII
NUMBER OF DUPLICATE TRACKS AS A FUNCTION OF $N$, THE EXPECTED NUMBER OF TARGET MEASUREMENTS PER SCAN

| | ML-PDA | | ML-PMHT | |
|---|---|---|---|---|
| N | mean | confidence interval (95 percent) | mean | confidence interval (95 percent) |
| 1 | 1.32 | [1.21, 1.43] | 0.03 | [0.00, 0.05] |
| 2 | 4.05 | [3.88, 4.23] | 0.04 | [0.01, 0.06] |
| 3 | 6.12 | [5.92, 6.31] | 0.09 | [0.05, 0.13] |

After this, ML–PDA and ML–PMHT were run on all five scenarios in their respective multitarget tracking modes. ML–PDA operated in the sequential single-target mode, while ML–PMHT was run in its true multitarget mode for tracks that were close together.

All the scenarios except one featured clutter that had $K$-distributed amplitudes. The high-clutter version of scenario 1 featured Rayleigh-distributed clutter amplitudes. In all cases, the feature LLRs calculated by ML–PDA and ML–PMHT matched the correct clutter distributions.

For scenarios with a single target or easily distinguished targets, ML–PDA and ML–PMHT had virtually identical performance over the Monte Carlo runs. As is shown in Section II-B, the ML–PMHT LLR should be very close to the ML–PDA LLR, so the tracking results of the two should be similar. The terms in (4) that represent more than one measurement in a scan originating from the target are clearly very close to zero, resulting in the ML–PDA and ML–PMHT LLRs being nearly identical and thus producing nearly identical tracking results.

Individual examples taken from the Monte Carlo runs help to further illustrate this. For scenario 1, an ML–PDA result shown in Fig. 5 and an ML–PMHT result shown in Fig. 6 are virtually indistinguishable from each other. For the same scenario with the Rayleigh clutter, results for the two algorithms (Fig. 15 for ML–PDA and Fig. 16 for ML–PMHT) are also very similar. Finally, for scenario 3, the ML–PDA example in Fig. 9 and the ML–PMHT example in Fig. 10 appear very much alike.

For the scenarios with targets in close proximity to each other with similar motion dynamics (scenarios 2 and 5), or the scenario with multiple targets that were close enough to potentially cause a tracker to switch targets (scenario 4), the results in Tables VI–XI show that ML–PMHT clearly outperforms ML–PDA. This is due to the true multitarget implementation for ML–PMHT in contrast to the sequential single-target tracking logic that is necessary for ML–PDA. In scenario 2, ML–PMHT is able to track targets 1 and 2 in excess of 80% of

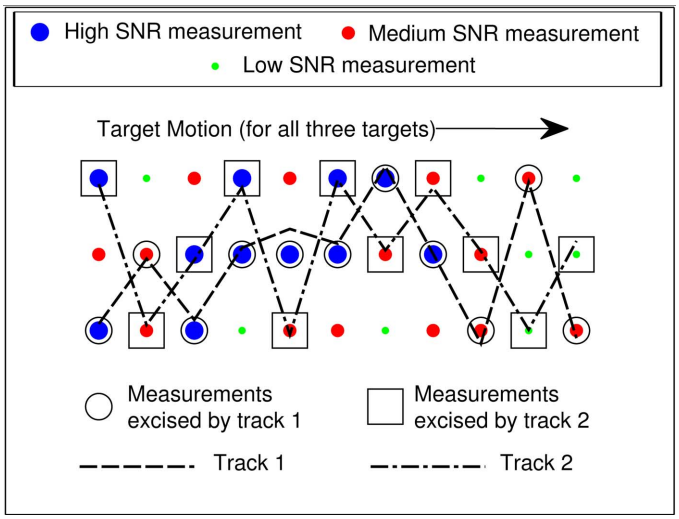


Fig. 17. Example of multitrack measurement assignment for ML–PDA.

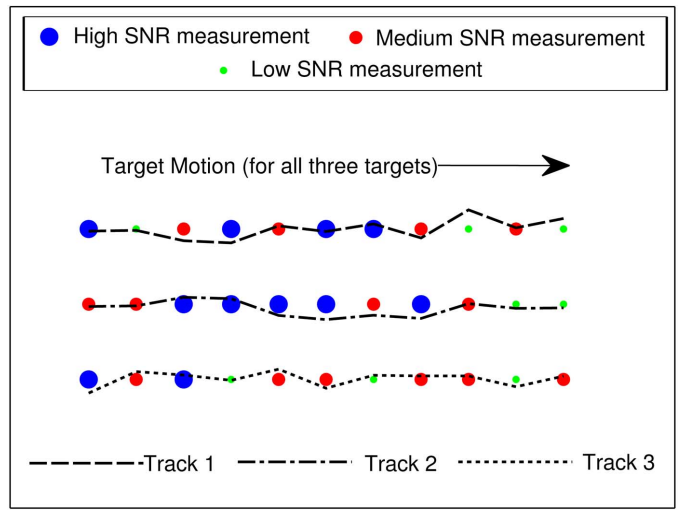


Fig. 18. Example of multitrack measurement assignment for ML–PMHT.

the time and track target 3 nearly 50% of the time. In contrast, ML–PDA is tracking target 2 (the middle target) 100% of the time, and targets 1 and 3 (the outer targets) less than 10% of the time. (Overall, ML–PMHT is tracking at least one target 100% of the time.)

The reason ML–PDA is not performing as well as ML–PMHT in this case is the sequential single-target implementation that is necessary for ML–PDA. Fig. 17 illustrates what is happening in scenario 2. This figure shows measurements from three targets in close proximity to each other, all moving in the same direction. For clarity of illustration, the measurements have no noise, and there are no clutter measurements shown. ML–PDA is implemented in a sequential mode; the first track "finds" the high-SNR measurement in each scan, and then these measurements are excised from the data. The next track finds the remaining (relatively) high-SNR measurement, and again, these measurements are excised from the data. Since both found tracks are using data from all three true targets, the tracks tend to zigzag back and forth over the middle target (and each other). As a result, the track scoring will most likely result in the middle target being tracked by both tracks. Over many trials, this will produce a high $P_d$ value for the middle target, a duplicate track for the middle target, and low $P_d$ for the outer targets. This is exactly what is seen in the Monte Carlo results for scenario 2: for ML–PDA the middle target had a $P_d$ of 100%, while the outer targets had a $P_d$ of less than 10%. The middle target also had, on average, 1.07 duplicate tracks associated with it.

In contrast, ML–PMHT, with its more natural and appealing true multitarget formulation, can better find all three targets. Simultaneously optimizing for multiple targets at once prevents the "claiming" of the high-SNR measurement by the first track to run through the data; instead, the high-SNR measurements are more equally (and correctly) divided between the three tracks, as is shown in Fig. 18. Scenario 5 showed similar results. ML–PDA was only able to track both targets approximately 45% of the time, while ML–PMHT was able to track both targets approximately 65% of the time. This number is actually slightly misleading; the relatively low numbers for ML–PDA are because tracks using this algorithm switched targets on the portion of the trajectory where the targets were close, and the tracks ended up on the wrong targets, driving down $P_d$. There was some switching for ML–PMHT as well, but as is seen by the results, ML–PDA was more susceptible to this problem. This is a result of the sequential single-target implementation used for ML–PDA that is described above.

Individual examples from the Monte Carlo runs again reinforce this conclusion. A scenario 2 result for ML–PDA is shown in Fig. 7. Here, the first track (on the middle target) "claims" and then excises all the high-SNR measurements. A second track is initiated, and it claims the remaining (relatively) high-SNR measurements, in the process crossing over the first track several times. As a result, it appears that this track would be associated with the middle target as well. There are not enough high-SNR measurements left to form a third track. In contrast, the multitarget ML–PMHT tracker, shown in Fig. 8, is able to track all three targets with a minimum of switching. Examples from scenario 5 show similar results. In Fig. 13, ML–PDA is not able to separate the targets on the portion of the track where the two targets are paralleling each other at very close range; during this portion, the tracks actually switch between targets five times. In contrast, in Fig. 14, the multitarget ML–PMHT tracker is able to track both targets without any switching.

Finally, ML–PMHT outperformed ML–PDA on scenario 4 for almost all of the targets for similar reasons. (Results from only targets 7 and 12 are shown below; they are fairly representative of all 13 targets in this scenario). Again, the sequential single-target tracking framework for ML–PDA could not perform as well as the true multitarget implementation for ML–PMHT. Tracks using the ML–PDA implementation were far more susceptible to being drawn off from one target to another by high-SNR measurements (similar to the effect illustrated in Fig. 17). Compare the performance of ML–PDA on this scenario in Fig. 11 with the performance of ML–PMHT in Fig. 12. Several instances of track switching are visible in the ML–PDA plot that are not seen in the ML–PMHT plot. While this switching is not between targets with similar dynamics (as happens with scenarios 2 and 5), it is happening with targets that

are moving very slowly. As a result, there are no dynamics—either evolving position over time or Doppler—to differentiate measurements between two targets.

All the simulations up to this point were performed using the ML–PDA target measurement generation model, that is, at most one measurement was generated by the target in any given scan. In this condition, ML–PMHT and ML–PDA had identical performance in the single-target cases, and ML–PMHT outperformed ML–PDA in the small-separation multitarget cases. We now round out the comparison between the two algorithms by considering what happens when the ML–PMHT target measurement generation model is used to generate the data and more than one measurement is allowed to originate from the target.

To do this, scenario 1 (with $K$-distributed clutter) was rerun for one, two, and three expected target returns per scan. Results for all metrics were virtually the same between the algorithms (and similar to results shown above for scenario 1), with the exception of the number of duplicate tracks. These results are shown in Table XII. Here, ML–PDA, as expected, suffers from an increasing number of duplicate tracks as $N$, the expected number of target measurements per scan, is increased. In contrast, ML–PMHT basically has on average no duplicate tracks for any number of target measurements per scan. With this model of target-measurement generation, ML–PMHT is the superior algorithm.

## VI. Conclusion

We developed a true multitarget implementation of ML–PMHT. In the process, we developed an expression for the ML–PMHT CRLB. We showed that ML–PMHT is a statistically efficient estimator, and that the ML–PMHT CRLB can be evaluated in real time, a valuable result in and of itself; this allows for a "complete" tracker framework that provides both a state estimate as well as a reliable estimate of the state covariance.

Then, we tested these developments by comparing ML–PDA and ML–PMHT with Monte Carlo testing. This testing first showed that ML–PDA and ML–PMHT are effective VLO trackers, working down to an expected target SNR of 4–5 dB (postsignal processing). After this, the ML–PDA and ML–PMHT tracking algorithms were applied to five different benchmark scenarios with Monte Carlo trials using a target measurement generation model of zero or one measurements originating from the target in a scan. For scenarios with a single target or multiple targets with measurements that could easily be differentiated by dynamics, the performances of ML–PDA and ML–PMHT were identical: ML–PMHT did not suffer from the fact that its measurement assignment model did not match the actual target measurement generation model. In cases with closely spaced targets with measurements that could not be differentiated easily by dynamics, ML–PMHT outperformed ML–PDA because the former had a true multitarget LLR formulation, while the latter had to handle multiple targets in a sequential single-target mode. Finally, when the target measurement generation model was switched to that of ML–PMHT, with multiple measurements per scan being generated by the target, ML–PMHT outperformed ML–PDA in terms of the number of duplicate tracks generated. Overall, the performance of ML–PMHT makes it the preferred algorithm.

## Appendix
## Calculation of the ML–PMHT FIM

Here, we fully derive the FIM for a window of ML–PMHT data. This is a specific implementation of the general work given in [22]. The derivation is also similar to that done for ML–PDA in [8]. However, the ML–PMHT calculation ends up being at most a fourfold integral (as opposed to approximately an $m$-fold integral for ML–PDA, where $m$ is the number of measurements in a scan), so this can be done in real time instead of having to calculate it offline with Monte Carlo integration, as must be done with ML–PDA.

Consider a window of ML–PMHT data with $N_w$ scans. Since all measurements from scan to scan are assumed to be independent, the FIM $\mathbf{J}$ of the total window can just be written as the sum of the FIM's $\mathbf{J}_i$ of the individual scans

$$\mathbf{J} = \sum_{i=1}^{N_w} \mathbf{J}_i. \tag{29}$$

Now, the FIM of a scan is calculated as

$$\mathbf{J}_i = E\{[\nabla_x \ln p[Z(i)\,|\,\mathbf{x}]][\nabla_x \ln p[Z(i)\,|\,\mathbf{x}]]^T\}|_{\mathbf{x}=\mathbf{x}_{\text{true}}}. \tag{30}$$

The ML–PMHT likelihood for a single scan is

$$p[Z(i)\,|\,\mathbf{x}] = \prod_{j=1}^{m_i}\left\{\frac{\pi_0}{V}p_0^\tau(a_j) + \pi_1 p[\mathbf{z}_j(i)\,|\,\mathbf{x}]p_1^\tau(a_j)\right\}. \tag{31}$$

Taking the gradient of the logarithm of this gives

$$\nabla_x \ln p[Z(i)\,|\,\mathbf{x}] = \sum_{j=1}^{m_i}\mathbf{D}_\phi^T \times \frac{\dfrac{\pi_1 p_1^\tau(a_j)}{\sqrt{|2\pi\mathbf{R}_j|}}e^{-\frac{1}{2}(\mathbf{z}_j-\boldsymbol{\phi})^T\mathbf{R}_j^{-1}(\mathbf{z}_j-\boldsymbol{\phi})}\mathbf{R}_j^{-1}(\mathbf{z}_j-\boldsymbol{\phi})}{\dfrac{\pi_0 p_0^\tau(a_j)}{V} + \dfrac{\pi_1 p_1^\tau(a_j)}{\sqrt{|2\pi\mathbf{R}_j|}}e^{-\frac{1}{2}(\mathbf{z}_j-\boldsymbol{\phi})^T\mathbf{R}_j^{-1}(\mathbf{z}_j-\boldsymbol{\phi})}}. \tag{32}$$

Here, $\phi$ is the state-to-measurement conversion and $\mathbf{D}_\phi$ is the Jacobian of $\phi$. (Note that we have dropped the scan index $i$ on the values $\mathbf{z}$, $\phi$, $\mathbf{R}$, and $a$ for ease of expression.) Inserting (32) into (30) and recognizing that cross terms in the product of the sums will go to zero results in (33), shown at the bottom of the page, where $d\mathbf{Z} = \prod_{l=1}^{m_i} d\mathbf{z}_l$ and $d\mathbf{a} = \prod_{l=1}^{m_i} da_l$. We are denoting the denominator in (32) as $p(\mathbf{z}_j\,|\,\mathbf{x})$. Note that all $p(\mathbf{z}_l\,|\,\mathbf{x})$ terms for $l \neq j$ are separable and simply integrate to

$$\mathbf{J}_i = \sum_{j=1}^{m_i}\int_V \mathbf{D}_\phi^T \frac{\dfrac{[\pi_1 p_1^\tau(a_j)]^2}{|2\pi\mathbf{R}_j|}e^{-(\mathbf{z}_j-\boldsymbol{\phi})^T\mathbf{R}_j^{-1}(\mathbf{z}_j-\boldsymbol{\phi})}\mathbf{R}_j^{-1}(\mathbf{z}_j-\boldsymbol{\phi})(\mathbf{z}_j-\boldsymbol{\phi})^T\mathbf{R}_j^{-1}}{p(\mathbf{z}_j\,|\,\mathbf{x})^2}\mathbf{D}_\phi \prod_{l=1}^{m_i} p(\mathbf{z}_l\,|\,\mathbf{x})d\mathbf{Z}d\mathbf{a} \tag{33}$$

$$\mathbf{J}_i = \sum_{j=1}^{m_i} \int_V \mathbf{D}_\phi^T \frac{\frac{[\pi_1 p_1^\tau(a_j)]^2}{|2\pi\mathbf{R}_j|} e^{-(\mathbf{z}_j-\boldsymbol{\phi})^T\mathbf{R}_j^{-1}(\mathbf{z}_j-\boldsymbol{\phi})}\mathbf{R}_j^{-1}(\mathbf{z}_j-\boldsymbol{\phi})(\mathbf{z}_j-\boldsymbol{\phi})^T\mathbf{R}_j^{-1}}{\frac{\pi_0 p_0^\tau(a_j)}{V} + \frac{\pi_1 p_1^\tau(a_j)}{\sqrt{|2\pi\mathbf{R}_j|}} e^{-\frac{1}{2}(\mathbf{z}_j-\boldsymbol{\phi})^T\mathbf{R}_j^{-1}(\mathbf{z}_j-\boldsymbol{\phi})}} \mathbf{D}_\phi d\mathbf{z}_j da_j \tag{34}$$

one, and the term for $l = j$ is canceled by one of the terms in the denominator. This happens because the likelihood function for ML–PMHT (for a scan) is a product, and this reduces the overall FIM integral from being an $m$-fold integral to only a threefold or fourfold integral. In contrast, as is shown in [8], the ML–PDA likelihood function for a scan of data consists of a summation, not a product, over all the measurements in the scan. As a result, taking the gradient of the logarithm of this results in summation over all the measurements in the denominator of the ML–PDA equivalent of (33). Thus, the cancellation that occurs above does not happen for ML–PDA; its FIM integral remains on the order of $m$-fold.

At this point, we can reduce (33) to (34), shown at the top of the page. The Jacobians are independent of the variables of integration and the summation indices, and thus can be moved outside the integral and summation. Furthermore, we can make the following substitution of:

$$\xi_j = \mathbf{G}_j(\mathbf{z}_j - \boldsymbol{\phi}) \tag{35}$$

where $\mathbf{G}_j$ is the Cholesky decomposition of $\mathbf{R}_j^{-1}$ such that $\mathbf{G}_j^T\mathbf{G}_j = \mathbf{R}_j^{-1}$. With this, it is possible to write the final expression for $\mathbf{J}_i$

$$\mathbf{J}_i = \mathbf{D}_\phi^T \sum_{j=1}^{m_i} \mathbf{G}_j^T \times \int_V \frac{\frac{[\pi_1 p_1^\tau(a_j)]^2}{|2\pi\mathbf{R}_j|} e^{-\xi_j^T\xi_j}\xi_j\xi_j^T}{\frac{\pi_0 p_0^\tau(a_j)}{V} + \frac{\pi_1 p_1^\tau(a_j)}{\sqrt{|2\pi\mathbf{R}_j|}} e^{-\frac{1}{2}\xi_j^T\xi_j}} d\xi_j da_j \frac{\mathbf{G}_j}{|\mathbf{G}_j|}\mathbf{D}_\phi. \tag{36}$$

This is a relatively simple threefold or fourfold integration (depending on the dimension of $\mathbf{z}_j$) in this application and can be done in real time. The resultant $\mathbf{J}_i$ for each scan is simply summed up then in accordance with (29) to get the FIM for a window of data.

## REFERENCES

[1] S. Schoenecker, P. Willett, and Y. Bar-Shalom, "A comparison of the ML-PDA and the ML-PMHT algorithms," in *Proc. 14th Int. Conf. Inf. Fusion*, Chicago, IL, USA, 2011, pp. 1–8.

[2] S. Schoenecker, P. Willett, and Y. Bar-Shalom, "Maximum likelihood probabilistic multi-hypothesis tracker applied to multistatic sonar data sets," in *Proc. SPIE Conf. Signal Process. Sensor Fusion Target Recognit.*, Orlando, FL, USA, 2011, DOI:10.1117/12.884766.

[3] C. Jauffret and Y. Bar-Shalom, "Track formation with bearing and frequency measurements in clutter," in *Proc. 29th Conf. Decision Control*, Honolulu, HI, USA, Dec. 1990, vol. 6, pp. 3335–3336.

[4] W. Blanding, P. Willett, and S. Coraluppi, "Sequential ML for multistatic sonar tracking," in *Proc. OCEANS Eur Conf.*, Aberdeen, Scotland, Jun. 2007, DOI: 10.1109/OCEANSE.2007.4302356.

[5] P. Willett and S. Coraluppi, "Application of the MLPDA to bistatic sonar," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2005, pp. 2063–2073.

[6] P. Willett and S. Coraluppi, "MLPDA and MLPMHT applied to some MSTWG data," in *Proc. 9th Int. Conf. Inf. Fusion*, Florence, Italy, Jul. 2006, DOI: 10.1109/ICIF.2006.301739.

[7] Y. Bar-Shalom, P. Willett, and X. Tian, "The ML-PDA estimator for VLO targets," in *Tracking and Data Fusion: A Handbook of Algorithms*. Bradford, U.K.: YBS Publishing, 2011, ch. 3.8.1 and 3.8.2, pp. 209–212.

[8] T. Kirubarajan and Y. Bar-Shalom, "Low observable target motion analysis using amplitude information," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 4, pp. 1637–1382, Oct. 1996.

[9] M. R. Chummun, Y. Bar-Shalom, and T. Kirubarajan, "Adaptive early-detection ML-PDA estimator for LO targets with EO sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 2, pp. 694–707, Apr. 2002.

[10] W. Blanding, P. Willett, and Y. Bar-Shalom, "ML-PDA: Advances and a new multitarget approach," *EURASIP J. Adv. Signal Process.*, vol. 2008, pp. 1–13, 2008.

[11] W. Blanding, P. Willett, and Y. Bar-Shalom, "Offline and real-time methods for ML-PDA track validation," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1994–2006, May 2007.

[12] R. Georgescu, P. Willett, and S. Schoenecker, "GM-CPHD and MLPDA applied to the SEABAR07 and TNO-blind multi-static sonar data," in *Proc. 12th Int. Conf. Inf. Fusion*, Seattle, WA, USA, 2009, pp. 1851–1858.

[13] R. Georgescu, P. Willett, and S. Schoenecker, "GM-CPHD and ML-PDA applied to the Metron multi-static sonar dataset," in *Proc. 13th Int. Conf. Inf. Fusion*, Edinburgh, Scotland, 2010, pp. 1–8.

[14] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[15] S. Coraluppi, "Mulitstatic sonar localization," *IEEE J. Ocean. Eng.*, vol. 31, no. 4, pp. 964–974, Oct. 2006.

[16] H. Cox, "Fundamentals of bistatic active sonar," BBN Syst. Technol. Corp., Cambridge, MA, Tech. Rep. W1038, 1988.

[17] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, "Definition and the statistical tests for filter consistency," in *Estimation With Applications to Tracking and Navigation*. New York, NY, USA: Wiley, 2001, ch. 5.4.2 and 5.4.3, pp. 234–240.

[18] P. Willett, Department of Electrical & Computer Engineering, University of Connecticut Website. Storrs, CT, USA, Aug. 2012 [Online]. Available: http://www.engr.uconn.edu/~willett/current_papers/

[19] D. Abraham, "Detection-threshold approximation for non-Gaussian backgrounds," *IEEE J. Ocean. Eng.*, vol. 35, no. 2, pp. 355–365, Apr. 2010.

[20] D. Abraham and A. Lyons, "Simulation of non-Rayleigh reverberation and clutter," *IEEE J. Ocean. Eng.*, vol. 29, no. 2, pp. 347–362, Apr. 2004.

[21] A. Abramowitz and I. Stegun, "Handbook of mathematical functions," in *National Bureau of Standards Applied Mathematics Series 55*. New York, NY, USA: Dover, 1965, ch. 9.1.1, 9.1.98, and 9.12.

[22] Y. Ruan, P. Willett, and R. Streit, "A comparison of the PMHT and PDAF tracking algorithms based on their model CRLBs," in *Proc. SPIE Aerosense Conf. Acquisit. Tracking Pointing*, Orlando, FL, USA, 1999, DOI:10.1117/12.352860.

**Steven Schoenecker** was born in 1972 in Washington, DC, USA. He received the A.B. degree in physics from Princeton University, Princeton, NJ, USA, in 1995 and the M.S. degree in computer science from the Rensselaer Polytechnic Institute at Hartford (Rensselaer at Hartford), Hartford, CT, USA, in 2002. Currently, he is working toward the Ph.D. degree in electrical engineering at the University of Connecticut, Storrs, CT, USA, working under the direction of Prof. P. Willett and Prof. Y. Bar-Shalom.

From 1995 to 2002, he was an officer in the U.S. Navy's submarine force. From 2002 to the present, he has worked at the Naval Undersea Warfare Center, Newport, RI, USA. His area of interest is statistical signal processing, with current emphasis on multitarget, multisensor tracking.

**Peter Willett** (S'83–M'86–SM'97–F'03) received the B.A.Sc. degree in engineering science from the University of Toronto, Toronto, ON, Canada, in 1982 and the Ph.D. degree from Princeton University, Princeton, NJ, USA, in 1986.

He has been a faculty member at the University of Connecticut, Storrs, CT, USA, ever since, and, since 1998, has been a Professor. His primary areas of research have been statistical signal processing, detection, machine learning, data fusion, and tracking. He has interests in and has published in the areas of change/abnormality detection, optical pattern recognition, communications, and industrial/security condition monitoring.

Dr. Willett was the Editor-in-Chief for the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS from 2006 to 2011, and is presently the Vice-President of Publications for the IEEE Aerospace and Electronic Systems Society (IEEE AESS). In the past, he has been an Associate Editor for three active journals: the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS (for data fusion and target tracking), the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS. He is also an Associate Editor for the IEEE AEROSPACE AND ELECTRONIC SYSTEMS MAGAZINE and ISIF's electronic *Journal of Advances in Information Fusion*. He was General Co-Chair (with S. Coraluppi) for the 2006 ISIF/IEEE Fusion Conference, Florence, Italy; Executive Chair of the 2008 Fusion Conference, Cologne, Germany; and Emeritus Chair for the 2011 Fusion Conference, Chicago, IL, USA. He was Program Co-Chair (with E. Santos) for the 2003 IEEE Conference on Systems, Man & Cybernetics, Washington, DC, USA; and Program Co-Chair (with P. Varshney) for the 1999 Fusion Conference, Sunnyvale, CA, USA. He has been a member of the IEEE Signal Processing Society's Sensor-Array & Multichannel (SAM) Technical Committee (TC) since 1997, and both serves on that TC's SAM conferences' program committees and maintains the SAM website.

**Yaakov Bar-Shalom** (S'63–M'66–SM'80–F'84) was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion—Israel Institute of Technology, Haifa, Israel, in 1963 and 1967, respectively, and the Ph.D. degree from Princeton University, Princeton, NJ, USA, in 1970, all in electrical engineering.

From 1970 to 1976, he was with Systems Control, Inc., Palo Alto, CA, USA. Currently, he is Board of Trustees Distinguished Professor in the Department of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut, Storrs, CT, USA. He is also Director of the Estimation and Signal Processing (ESP) Lab. His current research interests are in estimation theory, target tracking, and data fusion. He has published over 400 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (New York, NY, USA: Academic, 1988); the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Reading, MA, USA: Artech House, 1993) and *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (New York, NY, USA: Wiley, 2001); the advanced graduate texts *Multitarget-Multisensor Tracking: Principles and Techniques* (Bradford, U.K.: YBS Publishing, 1995) and *Tracking and Data Fusion* (Bradford, U.K.: YBS Publishing, 2011); and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Reading, MA, USA: Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000). He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas.

Prof. Bar-Shalom has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." During 1976 and 1977, he served as an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, and, from 1978 to 1981, as an Associate Editor of *Automatica*. He was Program Chairman of the 1982 American Control Conference (ACC), General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–1987, he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society (CSS), and, during 1987–1989, he was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (ISIF; 1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002, and Vice-President for Publications in 2004–2011. In 1987, he received the IEEE CSS Distinguished Member Award. Since 1995, he has been a Distinguished Lecturer of the IEEE Aerospace and Electronic Systems Society (IEEE AESS) and has given numerous keynote addresses at major national and international conferences. He is corecipient of the M. Barry Carlton Award for the best paper in the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS in 1995 and 2000, and recipient of the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002, he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008, he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and, in 2012, the Connecticut Medal of Technology. He is listed by academic.research.microsoft as first among the researchers in aerospace engineering based on the citations of his work.